

1

Introduction to Plugin Support

Chapter 1 - Introduction to Plugin Support

Plugin Support provides an interface for developers to integrate with VP-UML. Developers can develop their plugins for what they want. In this section, we will introduce the structure of a plugin.

plugin.xml

A plugin is defined in a XML file (**plugin.xml**). It includes the information (such as plugin id, provider, required libraries, etc...), custom actions (menu, toolbar and popup menu) and custom shapes/connector of the plugin.

For working with VP-UML in plugin, there are 4 main components must be known by developers: **Model, Diagram, Diagram Element and Action/Action Controller**.

Model

VP-UML provides the model elements which supported in UML 2.0/1.x (such as Use Case, Class, Lifeline, etc...).

Plugin allows developer to create, retrieve update and delete on the model elements on code. To access the model element classes, developer can either get the model through the popup menu context or through the project (to iterate through all the models).

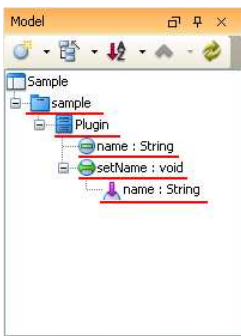


Figure 1.1 - Model Pane

The models are shown on the **Model** pane. There are **package** (*sample*), **class** (*Plugin*), **attribute** (*name*), **operation** (*setName*) and **parameter** (*name*) contained in the project.

Diagram

Diagram is the diagram which contains many diagram elements on different domain (such as Use Case Diagram, Class Diagram, ERD, etc...).

Plugin allows developer to create, retrieve, update and delete the diagrams on code. To access the diagrams, developer can either get the diagram through the popup menu context or through the project (to iterate through all the diagrams)

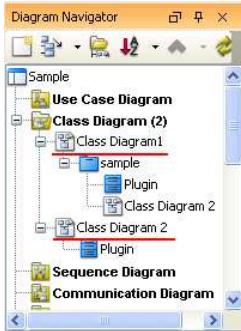


Figure 1.2 - Diagram Pane

The diagrams are shown on the **Diagram Navigator** pane. 2 Class Diagram are contains in the project (*Class Diagram 1* and *Class Diagram 2*).

Diagram Element

Diagram Element represents a model on a diagram. A model does not contain the information on outlook (such as x, y, width, height, etc...), and a model can be shown on different diagrams (such as a class can be shown on 2 different class diagrams). Diagram Elements work as an UI to show the model on different views for the user.

There are 2 kinds of diagram element: **Shape** and **Connector**.

Shape represents the non-relationships model (such as Class).

Connector represents the relationships model (such as Generalization).

Plugin allows developer to create, retrieve, update and delete the diagram elements on code. To access the diagram elements, developer can either get the diagram element through the popup menu context or through the project (to iterate through all the diagrams and iterate the diagram elements held on the diagram).

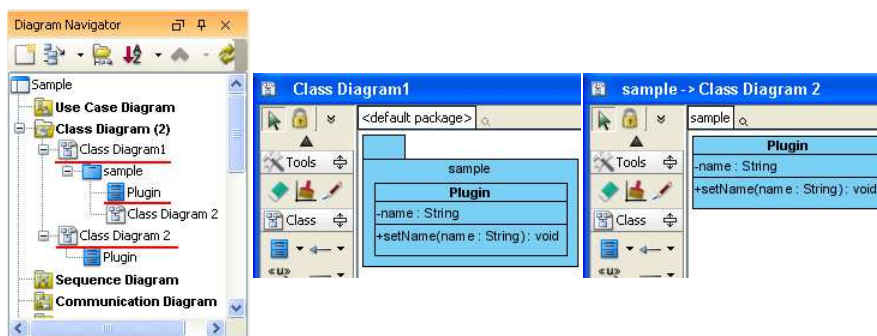


Figure 1.3 - Diagram element

The diagram elements are shown on the **Diagram Navigator** pane under the diagrams. The shapes are presenting the same model (*sample* and *Plugin*). And the shapes are shown on the 2 diagrams (*Class Diagram 1*, and *Class Diagram 2*) with same model, different outlook.

Action/Action Controller

Action represents the buttons (menu, toolbar and popup menu), which contains the information on outlook (such as label, icon, mnemonic, etc...) and responses to trigger the function call.

Action is used to represent the button on 3 regions: **menu/toolbar**, **popup menu** and **diagram toolbar**

Action Controller is the control (function call) of actions. Developer need to implement different Action Controller on different region's actions.

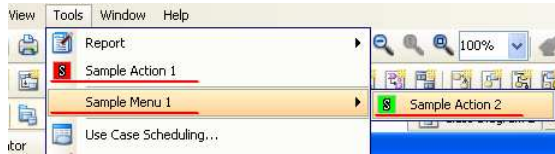


Figure 1.4 - Menu



Figure 1.5 - Toolbar

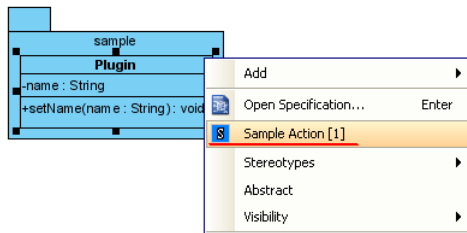


Figure 1.6 - Popup Menu

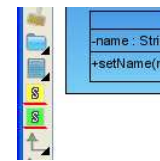


Figure 1.7 - Diagram Toolbar